

MENU

SEARCH

INDEX

DETAIL

JAPANESE

NEXT

1 / 4

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2004-004341
 (43)Date of publication of application : 08.01.2004

(51)Int.Cl. G09C 1/00

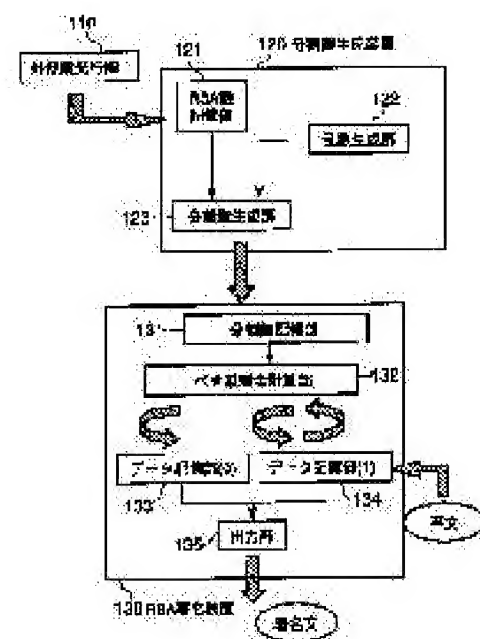
(21)Application number : 2002-160231 (71) TOSHIBA CORP
 Applicant :
 (22)Date of filing : 31.05.2002 (72)Inventor : TOMOE HIROKI
 KAWAMURA SHINICHI
 SHINPO ATSUSHI

(54) APPARATUS AND METHOD FOR MODULO EXPONENTIATION CALCULATION, AND PROGRAM

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a modulo exponentiation calculation apparatus which realizes prevention of secret key leakage due to power analysis attack to a public key ciphering system.

SOLUTION: A divided key generation part 120 generates k-number of divided secret keys from the secret key (d) of the public key ciphering system and generates x-number of dummy keys by a random number and a key index consisting of k-number of 1 and x-number of 0. An RSA signature device 130 writes a plaintext M as an initial value in data storage part (0), (1), performs modulo exponentiation calculation by using corresponding single divided secret key with the value of the data storage part (1) as an input when the key index is 1, stores the result in the data storage part (1), performs modulo exponentiation calculation by using corresponding single dummy key with the value of the data storage part (0) as an input when the key index is 0, and stores the result in the data storage part (0). The value of the data storage part (1) after processing is a signature sentence. Even if the divided secret keys are leaked due to power analysis attack, it is difficult to obtain the secret key (d).



LEGAL STATUS

[Date of request for examination] 17.09.2003

[Date of sending the examiner's decision of rejection] 03.10.2006

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号
特開2004-4341
(P2004-4341A)

(43) 公開日 平成16年1月8日(2004.1.8)

(51) Int.Cl.⁷
G09C 1/00

F I
G09C 1/00 650A
G09C 1/00 620A

テーマコード(参考)
5J104

審査請求 有 請求項の数 23 O L (全 29 頁)

(21) 出願番号	特願2002-160231 (P2002-160231)	(71) 出願人	000003078
(22) 出願日	平成14年5月31日 (2002.5.31)		株式会社東芝
			東京都港区芝浦一丁目1番1号
		(74) 代理人	100058479
			弁理士 鈴江 武彦
		(74) 代理人	100084618
			弁理士 村松 貞男
		(74) 代理人	100068814
			弁理士 坪井 淳
		(74) 代理人	100092196
			弁理士 橋本 良郎
		(74) 代理人	100091351
			弁理士 河野 哲
		(74) 代理人	100088683
			弁理士 中村 誠

最終頁に続く

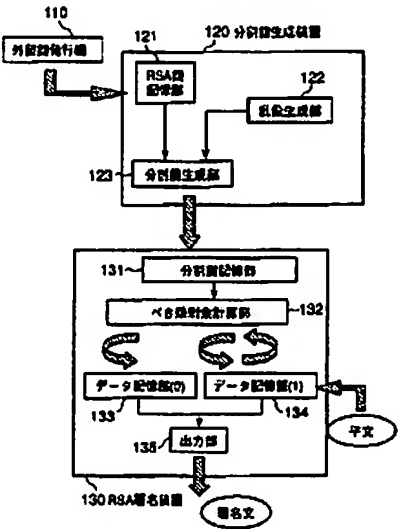
(54) 【発明の名称】 べき乗剰余計算装置、べき乗剰余計算方法及びプログラム

(57) 【要約】

【課題】 公開鍵暗号方式に対する電力解析攻撃による秘密鍵漏洩を防ぐことを可能にしたべき乗剰余計算装置を提供すること。

【解決手段】 分割鍵生成部120は、公開鍵暗号方式の秘密鍵dからk個の分割秘密鍵を生成し、乱数によりx個のダミー鍵、k個の1とx個の0からなる鍵インデックスを生成する。RSA署名装置130は、データ記憶部(0)、(1)に初期値として平文Mを書き込み、鍵インデックスが1ならばデータ記憶部(1)の値を入力として該当する1つの分割秘密鍵を用いてべき乗剰余算を行い、その結果をデータ記憶部(1)に格納し、0ならばデータ記憶部(0)の値を入力として該当する1つのダミー鍵を用いてべき乗剰余算を行い、その結果をデータ記憶部(0)に格納する。処理後のデータ記憶部(1)の値が署名文である。電力解析攻撃により分割秘密鍵が漏れても秘密鍵dを求めることは困難になる。

【選択図】 図1



【特許請求の範囲】

【請求項 1】

公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置において、

前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、

前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行う第 1 の計算手段とを備え、

前記第 1 の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他のべき乗剰余算の計算結果を入力とするものであり、

10

前記入力データに対するべき乗剰余は、前記第 1 の計算手段により行われた特定の一つの前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であることを特徴とするべき乗剰余計算装置。

【請求項 2】

前記秘密鍵をもとに前記複数の分割秘密鍵を生成する手段を更に備えたことを特徴とする請求項 1 に記載のべき乗剰余計算装置。

【請求項 3】

複数のダミー鍵を記憶する手段と、

前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行う第 2 の計算手段と、

前記第 1 の計算手段と前記第 2 の計算手段とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を記憶する手段とを更に備えたことを特徴とする請求項 1 に記載のべき乗剰余計算装置。

20

【請求項 4】

前記秘密鍵をもとに前記複数の分割秘密鍵を生成する手段と、

前記複数のダミー鍵を生成する手段と、

前記インデックス情報を生成する手段とを更に備えたことを特徴とする請求項 3 に記載のべき乗剰余計算装置。

【請求項 5】

前記第 1 の計算手段及び前記第 2 の計算手段は、それぞれ、前記インデックス情報の当該回に対応するフラグによる指示に従って、該当する回のみにおいて前記所定のべき乗剰余算を順次行うことを特徴とする請求項 3 または 4 に記載のべき乗剰余計算装置。

30

【請求項 6】

前記第 2 の計算手段における前記複数のダミー鍵の使用順序と、前記インデックス情報との少なくとも一方を、所定のタイミングで変更することを特徴とする請求項 3 ないし 5 のいずれか 1 項に記載のべき乗剰余計算装置。

【請求項 7】

前記第 1 の計算手段における前記複数の分割秘密鍵の使用順序と、前記所定の分割方法との少なくとも一方を、所定のタイミングで変更することを特徴とする請求項 1 ないし 6 のいずれか 1 項に記載のべき乗剰余計算装置。

【請求項 8】

40

公開鍵暗号方式の秘密鍵から派生した第 1 及び第 2 の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置において、

前記第 1 の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、

前記複数の第 1 の分割秘密鍵をそれぞれ用いた所定の第 1 のべき乗剰余算を順次行う第 1 の計算手段と、

前記第 2 の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、

前記複数の第 2 の分割秘密鍵をそれぞれ用いた所定の第 2 のべき乗剰余算を順次行う第 2 の計算手段と、

50

前記第 1 の計算手段と前記第 2 の計算手段とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を記憶するインデックス情報生成手段と

、
前記第 1 の計算手段による最終結果と前記第 2 の計算手段による最終結果とに基づいて、
前記入力データに対するべき乗剰余を求める手段とを備え、
前記第 1 の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他の第 1 のべき乗剰余算の計算結果を入力とするものであり、
前記第 2 の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他の第 2 のべき乗剰余算の計算結果を入力とするものであり、
前記第 1 の計算手段による最終結果は、前記第 1 の計算手段により行われた特定の一つの
前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に
基づいて得られる値であり、
前記第 2 の計算手段による最終結果は、前記第 2 の計算手段により行われた特定の一つの
前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に
基づいて得られる値であることを特徴とするべき乗剰余計算装置。

【請求項 9】

前記第 1 の秘密鍵をもとに前記複数の第 1 の分割秘密鍵を生成する手段と、
前記第 2 の秘密鍵をもとに前記複数の第 2 の分割秘密鍵を生成する手段とを更に備えたこと
を特徴とする請求項 8 に記載のべき乗剰余計算装置。

【請求項 10】

前記第 1 の計算手段及び前記第 2 の計算手段は、それぞれ、前記インデックス情報の当該
回に対応するフラグによる指示に従って、該当する回のみにおいて前記所定のべき乗剰余
算を順次行うことを特徴とする請求項 8 または 9 に記載のべき乗剰余計算装置。

【請求項 11】

複数のダミー鍵を記憶する手段と、
前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行う第 3 の計算手段とを
更に備え、
前記インデックス情報は、前記第 1 の計算手段乃至前記第 3 の計算手段のいずれによる前
記べき乗剰余算を行うかを指示するフラグの系列からなることを特徴とする請求項 8 に記
載のべき乗剰余計算装置。

【請求項 12】

前記第 1 の秘密鍵をもとに前記複数の第 1 の分割秘密鍵を生成する手段と、
前記第 2 の秘密鍵をもとに前記複数の第 2 の分割秘密鍵を生成する手段と、
前記複数のダミー鍵を生成する手段とを更に備えたことを特徴とする請求項 11 に記載の
べき乗剰余計算装置。

【請求項 13】

前記第 1 の計算手段乃至前記第 3 の計算手段は、それぞれ、前記インデックス情報の当該
回に対応するフラグによる指示に従って、該当する回のみにおいて前記所定のべき乗剰余
算を順次行うことを特徴とする請求項 11 または 12 に記載のべき乗剰余計算装置。

【請求項 14】

前記第 1 の計算手段における前記複数のダミー鍵の使用順序を所定のタイミングで変更す
ることを特徴とする請求項 11 ないし 13 のいずれか 1 項に記載のべき乗剰余計算装置。

【請求項 15】

前記第 1 の計算手段における前記複数の第 1 の分割秘密鍵の使用順序と、前記第 2 の計算
手段における前記複数の第 2 の分割秘密鍵の使用順序と、前記インデックス情報と、前記
所定の分割方法との少なくとも一つを、所定のタイミングで変更することを特徴とする請
求項 8 ないし 14 のいずれか 1 項に記載のべき乗剰余計算装置。

【請求項 16】

公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計
算方法において、

10

20

30

40

50

前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割するステップと、
前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、
少なくとも一つの前記所定のべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とするべき乗剰余計算方法。

【請求項 17】

公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、

前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割するステップと、

10

複数のダミー鍵を生成するステップと、

前記秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、

前記インデックス情報に従って、前記複数の秘密鍵及び前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、

少なくとも一つの前記秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とするべき乗剰余計算方法。

【請求項 18】

公開鍵暗号方式の秘密鍵から派生した第1及び第2の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、

20

前記第1の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第1の分割秘密鍵に分割するステップと、

前記第2の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第2の分割秘密鍵に分割するステップと、

前記第1の秘密鍵と前記第2の秘密鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、

前記インデックス情報に従って、前記複数の第1の秘密鍵及び前記複数の第2の秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、

少なくとも一つの前記第1の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第2の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とするべき乗剰余計算方法。

30

【請求項 19】

公開鍵暗号方式の秘密鍵から派生した第1及び第2の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、

前記第1の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第1の分割秘密鍵に分割するステップと、

前記第2の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第2の分割秘密鍵に分割するステップと、

複数のダミー鍵を生成するステップと、

40

前記第1の秘密鍵と前記第2の秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、

前記インデックス情報に従って、前記複数の第1の秘密鍵及び前記複数の第2の秘密鍵並びに前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、

少なくとも一つの前記第1の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第2の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とするべき乗剰余計算方法。

【請求項 20】

公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置としてコンピュータを機能させるためのプログラムであって、

50

前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割する機能と、
前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行う機能と、
少なくとも一つの前記所定のべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求める機能とをコンピュータに実現させるためのプログラム。

【請求項 2 1】

公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置としてコンピュータを機能させるためのプログラムであって、
前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割する機能と、
複数のダミー鍵を生成する機能と、
前記秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成する機能と、
前記インデックス情報に従って、前記複数の秘密鍵及び前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行う機能と、
少なくとも一つの前記秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求める機能とをコンピュータに実現させるためのプログラム。

【請求項 2 2】

公開鍵暗号方式の秘密鍵から派生した第 1 及び第 2 の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置としてコンピュータを機能させるためのプログラムであって、
前記第 1 の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第 1 の分割秘密鍵に分割する機能と、
前記第 2 の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第 2 の分割秘密鍵に分割する機能と、
前記第 1 の秘密鍵と前記第 2 の秘密鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成する機能と、
前記インデックス情報に従って、前記複数の第 1 の秘密鍵及び前記複数の第 2 の秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行う機能と、
少なくとも一つの前記第 1 の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第 2 の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求める機能とをコンピュータに実現させるためのプログラム。

【請求項 2 3】

公開鍵暗号方式の秘密鍵から派生した第 1 及び第 2 の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置としてコンピュータを機能させるためのプログラムであって、
前記第 1 の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第 1 の分割秘密鍵に分割する機能と、
前記第 2 の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第 2 の分割秘密鍵に分割する機能と、
複数のダミー鍵を生成する機能と、
前記第 1 の秘密鍵と前記第 2 の秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成する機能と、
前記インデックス情報に従って、前記複数の第 1 の秘密鍵及び前記複数の第 2 の秘密鍵並びに前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行う機能と、
少なくとも一つの前記第 1 の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第 2 の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求める機能とを有することを機能とをコンピュータに実現させるためのプログラム。

【発明の詳細な説明】

【 0 0 0 1 】

10

20

30

40

50

【発明の属する技術分野】

本発明は、公開鍵暗号方式の秘密鍵を用いてべき乗剰余計算を行うべき乗剰余計算装置、べき乗剰余計算方法及びプログラムに関する。

【0002】

【従来の技術】

公開鍵暗号方式の中で、最も広く利用されているRSA暗号を例に挙げて説明する。

【0003】

RSA暗号では、最初に、2つの大きな素数 p 、 q を決定し、 $n = p * q$ 、 $ed = 1 \pmod{\text{lcm}(p-1)(q-1)}$ を満たす n 、 e 、 d を求めることによって、公開鍵 (e, n) および秘密鍵 (d, p, q) となる鍵の生成を行う。なお、本明細書では、 $=$ は、合同を表すものとする。

10

【0004】

次に、例えばRSA署名を行う場合は、署名したい文 M に対して（秘密鍵を用いて）、 $S = M^d \pmod n$ なる処理を行うことになる。

【0005】

ところで、 $S = M^d \pmod n$ をその通りに計算すると、 d は n と同程度に大きい（例えば512ビットや1024ビット程度）ため、べき乗に大変時間がかかってしまう。そこで、この計算には、通常、square-multiply methodが使われる。

【0006】

図12のフローチャートを参照しながら、square-multiply methodの流れを説明する。

20

【0007】

ここで、例えば署名したい文 $M = x$ および秘密鍵 $d = b$ として、 $x^b \pmod n$ を例にする。

【0008】

べき指数 b は、

$b[k-1], b[k-2], \dots, b[0]$

と2進展開する（ただし、 $b[k-1]$ がMSB、 $b[0]$ がLSBである）。

【0009】

初期値を $z = 1$ 、 $i = k-1$ とする（ステップS101）。以降は、繰り返し処理となる。

30

【0010】

すなわち、まず、 $z = z^2 \pmod n$ なる処理をする（ステップS102）。

【0011】

次いで、べき指数のビット $b[i]$ が1ならば（ステップS103）、 $z = z * x \pmod n$ を計算し（ステップS104）、 i をデクリメントする（ステップS105）。他方、べき指数のビット $b[i]$ が0ならば（ステップS103）、何もせずに、 i をデクリメントする（ステップS105）。

【0012】

そして、 $i < 0$ でないならば（ステップS106）、ステップS102に戻り、次のループの処理を行う。

40

【0013】

最終的に、ステップS106において、 $i < 0$ となったならば、処理を終了する。このときの z が、求める解すなわち $x^b \pmod n$ である。

【0014】

このようなアルゴリズムを利用することで、べき乗剰余算の処理時間を大幅に短くすることができる。

【0015】

さらに、中国人剰余定理（Chinese Remainder Theorem（以下、

50

CRTと略す))を利用して処理時間を短縮する方法もある。これは、例えばRSAの処理の場合では、べき乗剰余の法を p 、 q に分解して、それぞれを法とした計算処理を行い、最後に結果を合成するという方法である。

【0016】

まず、 $S = M^d \bmod n$ を、

$S_p = M^{d_p} \bmod p$

$S_q = M^{d_q} \bmod q$

の2つに分解する。

ここで、

$M_p = M \bmod p$

$M_q = M \bmod q$

$d_p = d \bmod (p-1)$

$d_q = d \bmod (q-1)$

である。

【0017】

しかして、 S_p と S_q をそれぞれ計算した後に、

$S = \{S_p * q * (q^{-1} \bmod p) + S_q * p * (p^{-1} \bmod q)\} \bmod n$

という処理をして合成することによって、 $S = M^d \bmod n$ の結果を得ることができる。

【0018】

この場合においても、 $S_p = M^{d_p} \bmod p$ の計算と、 $S_q = M^{d_q} \bmod q$ の計算のそれぞれについて、square-multiply methodが用いられる。

【0019】

なお、上記では、RSA署名を例にとって説明したが、RSA署名の検証、RSAの暗号化、RSAの復号など、べき乗剰余算を行う他の暗号処理についても同様である。

【0020】

【発明が解決しようとする課題】

ところで、上記のような従来の方でRSA暗号を実装すると、square-multiply methodの際に、べき指数の2進展開で、 $b[i]$ が0のときと1のときで処理が違うために(図12のS103での分岐を参照)、暗号処理中の装置の電力を詳細に測定することで、消費電力波形に1のときの処理と0のときの処理との違いが観測され、結果として暗号の秘密鍵の情報が漏洩してしまうという問題点がある。例えば、通常のRSAの処理では秘密鍵 d の値が、中国人剰余定理を用いたRSAの処理では秘密鍵 d に対応する d_p 、 d_q の値が、消費電力波形から分かってしまう。このような攻撃法は、電力解析攻撃と呼ばれる。

【0021】

本発明は、上記事情を考慮してなされたもので、公開鍵暗号方式に対する電力解析攻撃による秘密鍵漏洩を防ぐことを可能にしたべき乗剰余計算装置、べき乗剰余計算方法及びプログラムを提供することを目的とする。

【0022】

【課題を解決するための手段】

本発明は、公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置において、前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行う第1の計算手段とを備え、前記第1の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他のべき乗剰余算の計算結果を入力とするものであり、前記入力データに対するべき乗剰余は、前記第1の計算手段により行われた特定の1つの前記べき乗

10

20

30

40

50

剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であることを特徴とする。

【0023】

また、本発明は、公開鍵暗号方式の秘密鍵から派生した第1及び第2の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算装置において、前記第1の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、前記複数の第1の分割秘密鍵をそれぞれ用いた所定の第1のべき乗剰余算を順次行う第1の計算手段と、前記第2の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づき分割して得られた、複数の分割秘密鍵を記憶する手段と、前記複数の第2の分割秘密鍵をそれぞれ用いた所定の第2のべき乗剰余算を順次行う第2の計算手段と、前記第1の計算手段と前記第2の計算手段とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を記憶するインデックス情報生成手段と、前記第1の計算手段による最終結果と前記第2の計算手段による最終結果とに基づいて、前記入力データに対するべき乗剰余を求める手段とを備え、前記第1の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他の第1のべき乗剰余算の計算結果を入力とするものであり、前記第2の計算手段により行われる前記べき乗剰余算は、前記入力データ又は先行して行われた他の第2のべき乗剰余算の計算結果を入力とするものであり、前記第1の計算手段による最終結果は、前記第1の計算手段により行われた特定の1つの前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であり、前記第2の計算手段による最終結果は、前記第2の計算手段により行われた特定の1つの前記べき乗剰余算の計算結果が示す値、又は特定の複数の前記べき乗剰余算の計算結果に基づいて得られる値であることを特徴とする。

【0024】

また、本発明は、公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割するステップと、前記複数の分割秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、少なくとも一つの前記所定のべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とする。

【0025】

また、本発明は、公開鍵暗号方式の秘密鍵を用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、前記秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の分割秘密鍵に分割するステップと、複数のダミー鍵を生成するステップと、前記秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、前記インデックス情報に従って、前記複数の秘密鍵及び前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、少なくとも一つの前記秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とする。

【0026】

また、本発明は、公開鍵暗号方式の秘密鍵から派生した第1及び第2の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、前記第1の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第1の分割秘密鍵に分割するステップと、前記第2の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第2の分割秘密鍵に分割するステップと、前記第1の秘密鍵と前記第2の秘密鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、前記インデックス情報に従って、前記複数の第1の秘密鍵及び前記複数の第2の秘密鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、少なくと

も一つの前記第1の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第2の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とする。

【0027】

また、本発明は、公開鍵暗号方式の秘密鍵から派生した第1及び第2の秘密鍵をそれぞれ用いて入力データに対するべき乗剰余を計算するべき乗剰余計算方法において、前記第1の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第1の分割秘密鍵に分割するステップと、前記第2の秘密鍵を、加法、減法、乗法若しくは除法又はそれらの組合せによる所定の分割方法に基づいて、複数の第2の分割秘密鍵に分割するステップと、複数のダミー鍵を生成するステップと、前記第1の秘密鍵と前記第2の秘密鍵と前記ダミー鍵とのいずれによる前記べき乗剰余算を行うかを指示するフラグの系列からなるインデックス情報を生成するステップと、前記インデックス情報に従って、前記複数の第1の秘密鍵及び前記複数の第2の秘密鍵並びに前記複数のダミー鍵をそれぞれ用いた所定のべき乗剰余算を順次行うステップと、少なくとも一つの前記第1の秘密鍵を用いたべき乗剰余算の計算結果及び一つの前記第2の秘密鍵を用いたべき乗剰余算の計算結果に基づいて、前記入力データに対するべき乗剰余を求めるステップとを有することを特徴とする。

10

【0028】

なお、装置に係る本発明は方法に係る発明としても成立し、方法に係る本発明は装置に係る発明としても成立する。

20

また、装置または方法に係る本発明は、コンピュータに当該発明に相当する手順を実行させるための（あるいはコンピュータを当該発明に相当する手段として機能させるための、あるいはコンピュータに当該発明に相当する機能を実現させるための）プログラムとしても成立し、該プログラムを記録したコンピュータ読取り可能な記録媒体としても成立する。

【0029】

本発明では、公開鍵暗号方式の秘密鍵（例えば、中国人剰余定理が用いられる場合には該秘密鍵から各法 p 、 q に対応する2つの秘密鍵 d_p 、 d_q のそれぞれ）をさらに分割した分割秘密鍵を用いて処理をするため、分割された一つ一つの分割秘密鍵は消費電力から分かたれてしまっても、それらから秘密鍵を求めることが困難である。本発明によれば、計算量的安全性によって秘密鍵漏洩を防ぐことが可能になる。さらに、秘密鍵の分割の仕方として、加法、減法、乗法若しくは除法またはそれらを組み合わせたものというように、あらゆる組み合わせで構成することができ、また、分割数や分割する鍵長を可変にすることや、ダミー鍵を挿入するなど、状況に応じてセキュリティレベルを変えることができる。

30

【0030】

【発明の実施の形態】

以下、図面を参照しながら発明の実施の形態を説明する。

【0031】

以下では、公開鍵暗号方式のうちRSA暗号の場合を例にとって説明する。また、以下では、秘密鍵を用いたべき乗剰余計算としてRSA暗号の署名生成の場合を例にとって説明する。

40

【0032】

（第1の実施形態）

まず、本発明の第1の実施形態として、中国人剰余定理を使用しない場合の実施形態について説明する。

【0033】

図1に、本実施形態に係る暗号処理システムの構成例を示す。図1は、分割鍵生成装置とRSA署名装置（べき乗剰余計算装置）とが分離している分離型の構成をとる場合の一例である。

【0034】

50

図1に示されるように、本暗号処理システムは、外部鍵発行機110、分割鍵生成装置120、RSA署名装置130を備えている。分割鍵生成装置120は、RSA鍵記憶部121、乱数生成部122、分割鍵生成部123を含み、RSA署名装置130は、分割鍵記憶部131、べき乗剰余計算部132、第1のデータ記憶部（以下、データ記憶部（0）と記述する）133、第2のデータ記憶部（以下、データ記憶部（1）と記述する）134、出力部135を含む。

【0035】

外部鍵発行機110は、RSAの公開鍵（ e, n ）と秘密鍵（ d, p, q ）を生成し、これらを分割鍵生成装置120内のRSA鍵記憶部121に供給する。ただし、 p, q はそれぞれ大きな素数で、 $n = p * q$, $ed = 1 \pmod{lcm(p-1)(q-1)}$ を満たす。

【0036】

分割鍵生成装置120は、分割秘密鍵等を生成してRSA署名装置130に供給し、RSA署名装置130は、供給された分割秘密鍵等を用いて平文Mに対する署名文 $M^d \pmod n$ を求めるものである。

【0037】

まず、分割鍵生成装置120について説明する。

【0038】

RSA鍵記憶部121は、外部鍵発行機110から供給されるRSAの公開鍵（ e, n ）と秘密鍵（ d, p, q ）を保持し、必要に応じて分割鍵生成部123に公開鍵（ e, n ）と秘密鍵（ d, p, q ）を供給する。

【0039】

乱数生成部122は、任意のビット長の鍵およびビット列を任意の数だけ乱数により生成し、それらを分割鍵生成部123に供給する。

【0040】

分割鍵生成部123は、乱数生成部122から供給される分割秘密鍵の構成成分と、RSA鍵記憶部121から供給される秘密鍵（ d, p, q ）、公開鍵（ e, n ）とを用いて、分割秘密鍵を生成する。

【0041】

以下、中国人剰余定理を使用しない場合の分割秘密鍵生成手順について説明する。

【0042】

図2に、この場合の処理手順の一例を示す。

【0043】

ここでは、分割秘密鍵生成の一例として、秘密鍵 d を、乗法のみにより、 k 個に分割するものとする。すなわち、

$d = d[0] * d[1] * \dots * d[k-1] \pmod{lcm(p-1, q-1)}$
となるように、 d を $d[0], \dots, d[k-1]$ に分割するものとする。

【0044】

まず、乱数生成部122は、 s ビットの乱数を k 個生成し、それぞれ、分割秘密鍵 $d[1] \sim d[k-1]$ へ代入する（ステップS1）。なお、 s は、秘密鍵 d のビット長より小さいビット数（秘密鍵 d が例えば512ビットあるいは1024ビット程度の場合に分割秘密鍵 $d[1] \sim d[k-1]$ がそれぞれ例えば8ビット程度）にすると好ましい。

【0045】

また、乱数生成部122は、 s ビットの乱数を x 個生成し、それぞれ、ダミー鍵 $du[0] \sim du[x-1]$ へ代入する（ステップS2）。

【0046】

また、乱数生成部122は、 $k+x-1$ ビット長で且つハミング重みを $k-1$ とする鍵インデックス $t[i]$ （ $1 \leq i \leq k+x-1$ ）をランダムに生成する（ステップS3）。

【0047】

次いで、分割鍵生成部123は、 $t[i] = 1$ となる鍵（つまり、分割秘密鍵 $d[j]$ ）（

$1 \leq j \leq k-1$)) を掛け合わせて、

$D_0 = d[1] * d[2] * \dots * d[k-1]$

とし、RSA鍵記憶部121から供給される秘密鍵 d を用いて、

$d[0] = d * D_0^{-1} \bmod (lcm(p-1, q-1))$

とすることで、残り1つの分割秘密鍵 $d[0]$ を求める(ステップS4)。

また、 $t[0] = 1$ とする(ステップS4)。

【0048】

以上で、

分割秘密鍵 $d[j]$ ($0 \leq j \leq k-1$)、

ダミー鍵 $du[u]$ ($0 \leq u \leq x-1$)、

鍵インデックス $t[i]$ ($0 \leq i \leq k+x-1$)

が完成する。

【0049】

そして、分割鍵生成部123は、

$d[j]$ ($0 \leq j \leq k-1$)、

$du[u]$ ($0 \leq u \leq x-1$)、

$t[i]$ ($0 \leq i \leq k+x-1$)、

公開鍵 (e, n) 、

秘密鍵 (d, p, q)

をRSA署名装置130へ出力する(ステップS5)。

【0050】

なお、図2の手順においては、 $d[1]$ 、 $d[2]$ 、 \dots 、 $d[k-1]$ を全て生成した後
に $d[0]$ を求める必要があるが、それ以外は、 k 個の分割秘密鍵と x 個のダミー鍵と鍵
インデックスをどのような順番で求めても構わない。

【0051】

次に、RSA署名装置130について説明する。

【0052】

分割鍵記憶部131は、分割鍵生成装置120内の分割鍵生成部123から供給される分
割秘密鍵、ダミー鍵、鍵インデックス、公開鍵 (e, n) 、秘密鍵 (d, p, q) を保持
し、必要に応じてそれぞれをべき乗剰余計算部132に供給する。

【0053】

データ記憶部(0)133には、ダミー鍵を使用したべき乗剰余計算の計算結果(すなわ
ちダミーの計算結果) S' が格納され、データ記憶部(1)134には、分割秘密鍵を使
用したべき乗剰余計算の計算結果(すなわち正規の計算結果) S が格納される。

【0054】

べき乗剰余計算部132は、データ記憶部(1)134から前回までの計算結果 S (た
だし、初回は初期値 $S = M$ となる)を呼び出して、分割鍵記憶部131から供給される鍵
インデックス値 $t[i]$ が1ならば、分割秘密鍵 $d[j]$ を使って $S = S^{-1} d[j] \bmod n$ の計算を行い、その結果 S をデータ記憶部(1)134に格納し、他方、鍵イン
デックス値 $t[i]$ が0ならば、ダミー鍵 $du[u]$ を使って $S' = S^{-1} du[u] \bmod n$ の計算をし、その結果 S' をデータ記憶部(0)133に格納する。なお、鍵イン
デックス値 $t[i] = 0$ の場合に、 $S' = S^{-1} du[u] \bmod n$ 以外の計算を行う
構成も可能である。

【0055】

なお、ここでは、鍵インデックス値 $t[i] = 1$ の場合に、分割秘密鍵によるべき乗剰余
計算を行い、鍵インデックス値 $t[i] = 0$ の場合に、ダミー鍵によるべき乗剰余計算を
行うようにしているが、これに限定されるものではなく、例えば、鍵インデックス値 $t[i] = 0$ の場合に、分割秘密鍵によるべき乗剰余計算を行い、鍵インデックス値 $t[i] = 1$ の場合に、ダミー鍵によるべき乗剰余計算を行うようにしてもよい。

【0056】

10

20

30

40

50

出力部 135は、データ記憶部(1) 134の最終結果 $S (= M^d \bmod n)$ となる)を参照し、これを最終署名結果として出力する。

【0057】

以下、中国人剰余定理を使用しない場合の分割秘密鍵を用いた RSA 署名手順について説明する。

【0058】

図3に、この場合の処理手順の一例を示す。

【0059】

なお、ここでは、分割秘密鍵 $d[j]$ は、 j の値の小さい方から逐次使用し、ダミー鍵 $du[u]$ は、 u の値の小さい方から逐次使用するものとした例を示している。

10

【0060】

最初に、データ記憶部(0) 133およびデータ記憶部(1) 134にそれぞれ初期値として平文 M を書き込み、また、 $i = 0$ 、 $j = 0$ 、 $u = 0$ とする(ステップ S11)。なお、データ記憶部(0) 133については、平文 M 以外の値が初期値となっても構わない。

【0061】

以降は、 $k + x$ 回の繰り返し処理となる。

【0062】

まず、べき乗剰余計算部 132は、鍵インデックス値 $t[i]$ を参照し、 $t[i] = 1$ ならば(ステップ S12)、データ記憶部(1) 134から S を読み出し、分割秘密鍵 $d[j]$ を使って、 $S = S^d[j] \bmod n$ の計算を行い、その結果 S をデータ記憶部(1) 134に格納し(ステップ S13)、 j および i をそれぞれ1ずつインクリメントする(ステップ S14、S17)。

20

【0063】

他方、鍵インデックス値 $t[i]$ が0ならば(ステップ S12)、データ記憶部(1) 134から S を読み出し、ダミー鍵 $du[u]$ を使って、 $S' = S^{du[u]} \bmod n$ の計算を行い、その結果 S' をデータ記憶部(0) 133に格納し(ステップ S15)、 u および i をそれぞれ1ずつインクリメントする(ステップ S16、S17)。

【0064】

上記の2系統のうちの鍵インデックス値 $t[i]$ に応じたいずれかが行われた後に、 i の値を参照し、 i の値が分割秘密鍵とダミー鍵の総数 $k + x$ を超えていないなら(ステップ S18)、ステップ S12に戻り、次のループの処理を行う。

30

【0065】

最終的に、ステップ S18において、 $i > (k + x)$ となったならば、処理を終了する。このときにデータ記憶部(1) 134に格納されている S が、求める署名文すなわち $M^d \bmod n$ である。

【0066】

以上のように処理を行うことで、分割秘密鍵によるべき乗剰余計算とダミー鍵によるべき乗剰余計算とをランダムな順序に実行させて、電力解析攻撃を困難にさせ、かつ、ダミー鍵の影響を残さずに、正しい最終署名結果を得ることができる。

【0067】

40

ここで、具体例で説明すると、例えば、 $k = 8$ 、 $x = 4$ 、鍵インデックス = "101101110101" の場合、

(1回目) $S = M^d[0] \bmod n$
 (2回目) $S' = S^{du[0]} \bmod n$
 (3回目) $S = S^d[1] \bmod n$
 (4回目) $S = S^d[2] \bmod n$
 (5回目) $S' = S^{du[1]} \bmod n$
 (6回目) $S = S^d[3] \bmod n$
 (7回目) $S = S^d[4] \bmod n$
 (8回目) $S = S^d[5] \bmod n$

50

(9回目) $S' = S^{d_u[2]} \bmod n$
 (10回目) $S = S^{d[6]} \bmod n$
 (11回目) $S' = S^{d_u[3]} \bmod n$
 (12回目) $S = S^{d[7]} \bmod n$

のように各ループ処理が行われ、

最終的に得られた S

$$= (((((((M^{d[0]} \bmod n)^{d[1]} \bmod n)^{d[2]} \bmod n)^{d[3]} \bmod n)^{d[4]} \bmod n)^{d[5]} \bmod n)^{d[6]} \bmod n)^{d[7]} \bmod n$$

$$= M^{d[0]} \cdot d[1] \cdot d[2] \cdot d[3] \cdot d[4] \cdot d[5] \cdot d[6] \cdot d[7] \bmod n$$

$$= M^{(d[0] * d[1] * d[2] * d[3] * d[4] * d[5] * d[6] * d[7])} \bmod n$$

$$= M^{d} \bmod n$$
 となる。

【0068】

このように本実施形態においては、電力解析攻撃によってRSA署名処理中の消費電力波形から、たとえ分割秘密鍵とダミー鍵の全ての値が分かったとしても、ダミー鍵と分割秘密鍵との区別が特定されなければ、秘密鍵 d を特定することができない。したがって、秘密鍵を求めるために必要な総当り数（測定される波形の個数 $x + k$ から分割秘密鍵の個数 k を選び出す組合せ）を大きくすることで、計算量的に攻撃を困難にすることができる。これまで説明した構成例では、鍵の分割数 k を多くすればするほど、セキュリティ強度が上がる。また、ダミー鍵の数を多くすればするほど、セキュリティ強度が上がる。

【0069】

以下では、本実施形態の変形例について説明する。

【0070】

上記では、分割秘密鍵 $d[j]$ は、 j の値の小さい方から逐次使用し、ダミー鍵 $d_u[u]$ は、 u の値の小さい方から逐次使用するものとしたが、もちろん、これらとは異なる順番で鍵を使用してもよい。

【0071】

また、上記では、分割秘密鍵 $d[j]$ の使用順序およびダミー鍵 $d_u[u]$ の使用順序は固定のものであったが、分割秘密鍵の使用順序とダミー鍵の使用順序の一方又は双方を、所定のタイミングで（例えば、毎回（本例では、署名文を生成するごとに））、ランダムに変えるようにしてもよい。例えば、毎回、乱数に基づいて全鍵の使用順序を決定するようにしてもよいし、また、例えば、予め鍵の使用順序のパターンを複数パターン用意しておき、それらパターンのうちから使用するものを乱数に基づいて決定するようにしてもよい。このように鍵の使用順序をランダムにすることによって、さらにセキュリティ強度を上げることができる。

【0072】

また、上記では、秘密鍵 d の分割方法として、乗法のみのものであったが、他の分割方法、例えば、加法のみ、減法のみ、除法のみ、また、加法・減法・乗法・除法の適当な組合せも可能である（例えば、 $d = d[0] + d[1] + d[2] + d[3] + \dots$ 、 $d = d[0] - d[1] - d[2] - d[3] - \dots$ 、 $d = d[0] / d[1] / d[2] / d[3] / \dots$ ）。

【0073】

また、上記では、秘密鍵 d の分割方法は特定のものに固定されていたが、例えば、装置（あるいはユーザ）毎に秘密鍵の分割方法を変える（同一の装置（あるいはユーザ）では秘密鍵の分割方法は固定する）ようにしてもよいし、また、例えば、同一の装置（あるいはユーザ）についても、所定のタイミングで（例えば、毎回）、鍵の分割方法を変えるよう

20

30

40

50

にしてもよい。このように秘密鍵の分割方法をランダムにすることによって、さらにセキュリティ強度を上げることができる。

【0074】

なお、分割方法を変えるにあたっては、分割方法自体を異なるものに変えてもよい。また、分割方法自体を異なるものに変えるのではなく、例えば、j番目の分割鍵 $d[j]$ に対する演算記号が加減乗除のいずれになるかが予め定められている特定の分割方法を用いるものとして、この特定の分割方法は変えずに、秘密鍵の分割数のみを変えるようにしてもよい。例えば、上記の乗算のみの分割方法の例では、各分割鍵に対する演算記号は必ず乗算記号になるので、分割方法としては乗法のみを用いた分割方法を用いるものとして、秘密鍵 d の分割数 k を可変とする場合が該当する。これらのようにすることによって、さ

10

【0075】

ここで、乗法以外のみの分割方法の一例として、加法のみの分割方法の場合について、前述した乗法に関する構成例を修正する部分について説明する。

【0076】

まず、秘密鍵 d を加法により k 分割すると、分割鍵は次のようになる。

$$d = d[0] + d[1] + \dots + d[k-1] \quad \text{mod} \quad (\text{lcm}(p-1, q-1))$$

$$D_0 = d[1] + d[2] + \dots + d[k-1]$$

$$d[0] = d - D_0 \quad \text{mod} \quad (\text{lcm}(p-1, q-1))$$

なお、 $d[1], d[2] \dots d[k-1]$ は、乗法に関する構成例と同様、 s ビットの乱

20

【0077】

また、図3の処理手順例においては、ステップS11で S の初期値を1とし、ステップS13の $S = S^{\wedge} d[i] \quad \text{mod} \quad n$ を、 $S = S * M^{\wedge} d[i] \quad \text{mod} \quad n$ とすればよい。なお、ステップS15については、 $S = S * M^{\wedge} d[i] \quad \text{mod} \quad n$ に対応させて、 $S' = S * M^{\wedge} d[u] \quad \text{mod} \quad n$ とすればよい。

【0078】

ここで、具体例で説明すると、例えば、 $k=8$ 、 $x=4$ 、鍵インデックス="101101110101"の場合、

$$(1\text{回目}) \quad S = 1 * M^{\wedge} d[0] \quad \text{mod} \quad n$$

30

$$(2\text{回目}) \quad S' = S * M^{\wedge} d[u[0]] \quad \text{mod} \quad n$$

$$(3\text{回目}) \quad S = S * M^{\wedge} d[1] \quad \text{mod} \quad n$$

$$(4\text{回目}) \quad S = S * M^{\wedge} d[2] \quad \text{mod} \quad n$$

$$(5\text{回目}) \quad S' = S * M^{\wedge} d[u[1]] \quad \text{mod} \quad n$$

$$(6\text{回目}) \quad S = S * M^{\wedge} d[3] \quad \text{mod} \quad n$$

$$(7\text{回目}) \quad S = S * M^{\wedge} d[4] \quad \text{mod} \quad n$$

$$(8\text{回目}) \quad S = S * M^{\wedge} d[5] \quad \text{mod} \quad n$$

$$(9\text{回目}) \quad S' = S * M^{\wedge} d[u[2]] \quad \text{mod} \quad n$$

$$(10\text{回目}) \quad S = S * M^{\wedge} d[6] \quad \text{mod} \quad n$$

$$(11\text{回目}) \quad S' = S * M^{\wedge} d[u[3]] \quad \text{mod} \quad n$$

40

$$(12\text{回目}) \quad S = S * M^{\wedge} d[7] \quad \text{mod} \quad n$$

のように各ループ処理が行われ、

最終的に得られたS

$$\begin{aligned}
 &= ((((((1 * M^{d[0] \bmod n} * M^{d[1] \bmod n} * M^{d[2] \bmod n} \\
 &* M^{d[3] \bmod n} * M^{d[4] \bmod n} * M^{d[5] \bmod n} * M^{d[6] \bmod n} \\
 &* M^{d[7] \bmod n} \\
 &= M^{d[0] * M^{d[1] * M^{d[2] * M^{d[3] * M^{d[4] * M^{d[5] * M^{d[6] * M^{d[7] \bmod n}
 \end{aligned}$$

$$\begin{aligned}
 &= M^{(d[0]+d[1]+d[2]+d[3]+d[4]+d[5]+d[6]+d[7]) \bmod n} \\
 &= M^{d \bmod n}
 \end{aligned}$$

10

となる。

【0079】

また、他の分割方法の例として、乗算及び加法による分割方法の場合について、前述した乗法に関する構成例を修正する部分について説明する。

【0080】

まず、秘密鍵dを乗算及び加法によりk分割した一例を次に示す（ただし、加法はd[k/2-1]とd[k/2]の間の1つのみとする）。

$$d = d[0] * \dots * d[k/2 - 1] + d[k/2] * \dots * d[k - 1] \bmod (1 \text{ cm}(p - 1, q - 1)) \quad 20$$

$$D_0 = d[1] * \dots * d[k/2 - 1]$$

$$D_1 = d[k/2] * \dots * d[k - 1]$$

$$d[0] = (d - D_1) * D_0^{-1} \bmod (1 \text{ cm}(p - 1, q - 1))$$

なお、d[1]、d[2]…d[k-1]は、乗法に関する構成例と同様、sビットの乱数を代入すればよい。

【0081】

また、図3の処理手順例においては、S1とS2の2つのパラメータを用意し、ステップS11でS1およびS2の初期値をそれぞれMとし、ステップS13を、j=0~k/2-1について、S1=S1^d[i] mod n、j=k/2~k-2について、S2=S2^d[i] mod n、j=k-1について、S2=S2^d[i] mod nを行った後に、S=S1*S2 mod nとすればよい。なお、ステップS15については、例えば、S'=S1^du[u] mod nあるいはS'=S2^du[u] mod nとすればよいし、また、例えば、uの値に応じて、S'=S1^du[u] mod nとS'=S2^du[u] mod nのいずれを行うかを選択するようにしてもよい。

30

【0082】

ここで、具体例で説明すると、例えば、k=8、x=4、鍵インデックス="101101110101"の場合、

$$(1 \text{ 回目}) S1 = M^{d[0]} \bmod n$$

40

$$(2 \text{ 回目}) S' = S1^{du[0]} \bmod n$$

$$(3 \text{ 回目}) S1 = S1^{d[1]} \bmod n$$

$$(4 \text{ 回目}) S1 = S1^{d[2]} \bmod n$$

$$(5 \text{ 回目}) S' = S1^{du[1]} \bmod n$$

$$(6 \text{ 回目}) S1 = S1^{d[3]} \bmod n$$

$$(7 \text{ 回目}) S2 = M^{d[4]} \bmod n$$

$$(8 \text{ 回目}) S2 = S2^{d[5]} \bmod n$$

$$(9 \text{ 回目}) S' = S2^{du[2]} \bmod n$$

$$(10 \text{ 回目}) S2 = S2^{d[6]} \bmod n$$

$$(11 \text{ 回目}) S' = S2^{du[3]} \bmod n$$

50

(12回目) $S2 = S2^{d[7]} \bmod n$

のように各ループ処理が行われ、

最終的に得られたS

$= S1 * S2 \bmod n$

$= (((M^{d[0] \bmod n})^{d[1] \bmod n})^{d[2] \bmod n})^{d[3] \bmod n} *$

$((((M^{d[4] \bmod n})^{d[5] \bmod n})^{d[6] \bmod n})^{d[7] \bmod n}) \bmod n$

$= M^{d[0]*d[1]*d[2]*d[3]*d[4]*d[5]*d[6]*d[7]} \bmod n$

10

$= M^{(d[0]*d[1]*d[2]*d[3]+d[4]*d[5]*d[6]*d[7])} \bmod n$

$= M^d \bmod n$

となる。

【0083】

さらに、他の分割方法も可能であり、分割方法に応じて、 $d[0]$ や、図3の処理手順例のステップS13の処理が定義される。ただし、分割方法によっては、 $d[0]$ の値を求める過程で逆元計算（例えば、 D_0^{-1} ）が必要になる場合に、その逆元計算において逆元が存在しないために、 $d[0]$ の値を求めることができないときは、その分割方法を使用することはできない。したがって、逆元計算を要する分割方法においては、 $d[1]$ 、 $d[2] \dots d[k-1]$ から $d[0]$ を求めるにあたって、必要な逆元が存在するかどうかをチェックして、存在しなければ、もう一度乱数を発生させて、 $d[1]$ 、 $d[2] \dots d[k-1]$ （の全部又は一部）を求め直すようにしてもよい。また、複数の分割方法を用意しておき、必要な逆元が存在しなければ、分割方法自体を変えるようにしてもよい。

20

【0084】

ところで、図1は、分割鍵生成装置とRSA署名装置とが分離している分離型の構成をとる場合の一例であったが、分割鍵生成装置とRSA署名装置が一体化している一体型の構成をとることも可能である。図4に、この場合の暗号処理システムの構成例を示す。図4に示されるように、本暗号処理システムは、図1の外部鍵発行機110と同様の外部鍵発行機310と、図1の分割鍵生成装置120の構成要素とRSA署名装置130の構成要素を包含するRSA署名装置（べき乗剰余計算装置）330を備えている。なお、図4の構成例の動作は、基本的には、図1の構成例と同様である。

30

【0085】

前者の図1の構成例の場合には、例えば、鍵発行側が外部鍵発行機110及び分割鍵生成装置120を備え、各利用者が例えばICカードなどからなるRSA署名装置130を利用するような形態が考えられる。この場合には、例えば、まず、鍵発行側において外部鍵発行機110及び分割鍵生成装置120により、当該利用者に応じた各種鍵をRSA署名装置130に書き込み、該利用者は鍵発行側から該各種鍵を書き込まれたRSA署名装置130を取得し、これを計算機等の装置に装着するなどして使用する。

40

【0086】

この場合、RSA署名装置130に書き込まれた鍵は、固定、または半固定で使うことが想定される。半固定にする場合、鍵の変更は、鍵発行側に行って貰う。半固定にする場合の鍵変更時の処理（データのやり取り）の手段としては、例えば、RSA署名装置130を現実には鍵発行側へ渡して分割鍵生成装置120により鍵の変更（新たな分割秘密鍵の書き込み等）を行って貰う方法や、RSA署名装置130と分割鍵生成装置120との間をネットワークを介して接続可能にして鍵の変更（新たな分割秘密鍵の書き込み等）を行って貰う方法などがある。ネットワークを介する場合、RSA署名装置130と分割鍵生成装置120との間でのやり取りには、認証や暗号化などの技術を利用するのが望まし

50

い。

【0087】

なお、この利用形態の場合には、外部鍵発行機110及び分割鍵生成装置120を一体化して構成してもよい。

【0088】

後者の図4の構成例の場合には、例えば、鍵発行側が外部鍵発行機210を備え、各利用者が例えばICカードなどからなるRSA署名装置220を利用するような形態が考えられる。この場合には、例えば、鍵発行側において外部鍵発行機210により、当該利用者に応じた秘密鍵及び公開鍵をRSA署名装置220に書き込み、該利用者は鍵発行側から該秘密鍵及び公開鍵を書き込まれたRSA署名装置220を取得し、該RSA署名装置220これを計算機等の装置に装着するなどして使用する。なお、図1の構成例で、例えば、鍵発行側が外部鍵発行機110を備え、各利用者が例えばICカードなどからなる分割鍵生成装置120及びRSA署名装置130を利用するような場合も同様である。

10

【0089】

この場合には、鍵インデックス値、分割秘密鍵とダミー鍵の値、鍵の分割方法などを、例えば署名生成の処理毎など、上記の半固定の利用形態よりもきめ細かい単位毎に（タイミングで）、変更することができ、更なるセキュリティ強度の向上が期待される。

【0090】

なお、図1や図4とは異なる構成例も可能である。例えば、図1において、乱数生成部122をRSA署名装置130側に備える構成も可能である。

20

【0091】

ところで、これまで図1や図4を参照しながら説明してきた構成例において、秘密鍵の分割方法や分割秘密鍵の使用順序などの分割鍵に関する要素の少なくとも一部を可変として、ダミー鍵を使わずに、分割秘密鍵だけ用いて処理をするようにしてもよい。これによっても、秘密鍵の分割方法あるいは分割秘密鍵の使用順序などが不明であるという点で、セキュリティ強度を上げることができる。

【0092】

ダミー鍵を使用しない場合には、図2の分割秘密鍵生成手順については、ステップS2、S3を省き、ステップS4で $t[0]$ を求める部分を省き、ステップS5でダミー鍵 $du[u]$ と鍵インデックス $t[i]$ を出力する部分を省く修正をすればよい。また、図3のRSA署名手順は、図5のように修正すればよい。

30

【0093】

なお、ダミー鍵を使用する使用しないかを、所定のタイミングで（例えば、毎回）、選択できる構成にしてもよい。

【0094】

また、常にダミー鍵を使用しない構成にする場合には、図1や図4の乱数生成部については、ダミー鍵および鍵インデックスを生成する機能が不要になり、ダミーの計算結果 S' を保持するデータ記憶部(0)は、それ自体が不要になる。

【0095】

（第2の実施形態）

次に、本発明の第2の実施形態として、中国人剰余定理を使用する場合の実施形態について説明する。

40

【0096】

第1の実施形態では、 $M^d \bmod n$ の計算において、 d に分割秘密鍵の手法を適用したが、本実施形態では、 $M^d \bmod n$ の計算に中国人剰余定理を使用する場合の $S_p = M^p \cdot d_p \bmod p$ および $S_q = M^q \cdot d_q \bmod q$ における d_p と d_q の個々に分割秘密鍵の手法を適用するものである。したがって、本実施形態は、第1の実施形態を、中国人剰余定理を使用するように修正したものであり、分割秘密鍵の手法に関する部分は、基本的には、第1の実施形態と同様である（第1の実施形態の変形例の説明も、本実施形態について同様に当て嵌まる）。

50

【0097】

図6に、本実施形態に係る暗号処理システムの構成例を示す。図6は、分割鍵生成装置とRSA署名装置（べき乗剰余計算装置）とが分離している分離型の構成をとる場合の一例である。

【0098】

図6に示されるように、本暗号処理システムは、外部鍵発行機310、分割鍵生成装置320、RSA署名装置330を備えている。また、分割鍵生成装置320は、RSA鍵記憶部321、乱数生成部322、分割鍵生成部323を含む。RSA署名装置330は、分割鍵記憶部331、べき乗剰余計算部332、第1のデータ記憶部（以下、データ記憶部（0）と記述する）333、第2のデータ記憶部（以下、データ記憶部（1）と記述する）334、平文分割部335、CRT合成部336、出力部337を含む。

10

【0099】

外部鍵発行機310は、RSAの公開鍵（ e, n ）、秘密鍵（ d, dp, dq, p, q ）を生成し、分割鍵生成装置320内のRSA鍵記憶部321に供給する。ただし、 p, q はそれぞれ大きな素数で、 $n = p * q$, $ed = 1 \pmod{\text{lcm}(p-1, q-1)}$ を満たし、 $dp = d \pmod{p-1}$, $dq = d \pmod{q-1}$ である。

【0100】

分割鍵生成装置320は、 dp に対する分割秘密鍵及び dq に対する分割秘密鍵等を生成してRSA署名装置330に供給し、RSA署名装置330は、供給されたそれら分割秘密鍵等を用いて中国人剰余定理を使用して平文 M に対する署名文 $M^d \pmod{n}$ を求めるものである。

20

【0101】

まず、分割鍵生成装置320について説明する。

【0102】

RSA鍵記憶部321は、外部鍵発行機310から供給されるRSAの公開鍵（ e, n ）と秘密鍵（ d, dp, dq, p, q ）を保持し、必要に応じて分割鍵生成部324に公開鍵（ e, n ）と秘密鍵（ d, dp, dq, p, q ）を供給する。

【0103】

乱数生成部322は、任意のビット長の鍵およびビット列を任意の数だけ乱数により生成し、それらを分割鍵生成部323に供給する。

30

【0104】

分割鍵生成部323は、乱数生成部322から供給される法 p 上の分割秘密鍵の構成成分と法 q 上の分割秘密鍵の構成成分と、RSA鍵記憶部321から供給される秘密鍵（ d, dp, dq, p, q ）、公開鍵（ e, n ）を用いて法 p 上の分割秘密鍵と法 q 上の分割秘密鍵を生成する。

【0105】

以下、中国人剰余定理を使用する場合の分割秘密鍵生成手順について説明する。

【0106】

図7に、この場合の処理手順の一例を示す。

40

【0107】

ここでは、分割秘密鍵生成の一例として、秘密鍵 dp を乗法のみにより k_p 個に分割し、秘密鍵 dq を乗法のみにより k_q 個に分割するものとする。すなわち、

$$dp = dp[0] * dp[1] * \dots * dp[k_p - 1] \pmod{p-1},$$

$$dq = dq[0] * dq[1] * \dots * dq[k_q - 1] \pmod{q-1}$$

となるように、 dp を $dp[0], \dots, dp[k_p - 1]$ に分割し、 dq を $dq[0], \dots, dq[k_q - 1]$ に分割する。なお、以下では、 $k_p = k_q = k$ として説明する。

【0108】

まず、乱数生成部322は、 s ビットの乱数を $k_p - 1 = k - 1$ 個生成し、それぞれ、分割秘密鍵 $dp[1] \sim dp[k - 1]$ へ代入する（ステップS21）。なお、 s は、秘密

50

鍵 d のビット長より小さいビット数（秘密鍵 d が例えば 512 ビットあるいは 1024 ビット程度の場合に分割秘密鍵 $d_p[1] \sim d_p[k-1]$ がそれぞれ例えば 8 ビット程度）にすると好ましい。

【0109】

また、乱数生成部 322 は、 s ビットの乱数を $k_q - 1 = k - 1$ 個生成し、それぞれ、分割秘密鍵 $d_q[1] \sim d_q[k-1]$ へ代入する（ステップ S22）。

【0110】

また、乱数生成部 322 は、 $(k_p - 1) + (k_q - 1) = 2(k - 1)$ ビット長で且つハミング重みを $k_p - 1 = k - 1$ とする鍵インデックス $t[i]$ ($2 \leq i \leq 2k - 1$) をランダムに生成する（ステップ S23）。

10

【0111】

次いで、分割鍵生成部 323 は、 $t[i] = 1$ となる鍵（つまり、分割秘密鍵 $d_p[j]$ ($1 \leq j \leq k - 1$)）を掛け合わせて、

$$D_p = d_p[1] * d_p[2] * \dots * d_p[k-1]$$

とし、RSA 鍵記憶部 321 から供給される秘密鍵 d_p を用いて、

$$d_p[0] = d_p * D_p^{-1} \mod (p-1)$$

とすることで、法 p 上の残り 1 つの分割秘密鍵 $d_p[0]$ を求める（ステップ S24）。

【0112】

同様にして、分割鍵生成部 323 は、 $t[i] = 0$ となる鍵（つまり、分割秘密鍵 $d_q[j]$ ($1 \leq j \leq k - 1$)）を掛け合わせて、

20

$$D_q = d_q[1] * d_q[2] * \dots * d_q[k-1]$$

とし、RSA 鍵記憶部 321 から供給される秘密鍵 d_q を用いて、

$$d_q[0] = d_q * D_q^{-1} \mod (q-1)$$

とすることで、法 q 上の残り 1 つの分割秘密鍵 $d_q[0]$ を求める（ステップ S24）。

また、 $t[0] = 1$ 、 $t[1] = 0$ とする（ステップ S24）。なお、その逆に、 $t[0] = 0$ 、 $t[1] = 1$ としてもよい。

【0113】

以上で、

法 p 上の分割秘密鍵 $d_p[j]$ ($0 \leq j \leq k - 1$)、

法 q 上の分割秘密鍵 $d_q[u]$ ($0 \leq u \leq k - 1$)、

30

鍵インデックス $t[i]$ ($0 \leq i \leq 2k - 1$)

が完成する。

【0114】

そして、分割鍵生成部 323 は、

$d_p[j]$ ($0 \leq j \leq k - 1$)、

$d_q[u]$ ($0 \leq u \leq k - 1$)、

$t[i]$ ($0 \leq i \leq 2k - 1$)、

公開鍵 (e, n) 、

秘密鍵 (d, d_p, d_q, p, q)

を RSA 署名装置 330 へ出力する（ステップ S25）。

40

【0115】

なお、図 7 の手順においては、 $d_p[1]$ 、 $d_p[2]$ 、 \dots 、 $d_p[k-1]$ を全て生成した後に $d_p[0]$ を求める必要があり、 $d_q[1]$ 、 $d_q[2]$ 、 \dots 、 $d_q[k-1]$ を全て生成した後に $d_q[0]$ を求める必要があるが、それ以外は、法 p 上の k 個の分割秘密鍵と法 q 上の k 個の分割秘密鍵と鍵インデックスをどのような順番で求めても構わない。

【0116】

次に、RSA 署名装置 330 について説明する。

【0117】

分割鍵記憶部 331 は、分割鍵生成装置 320 内の分割鍵生成部 323 から供給される法

50

p 上の分割秘密鍵、法 q 上の分割秘密鍵、鍵インデックス、公開鍵 (e, n)、秘密鍵 (d, d_p, d_q, p, q) を保持し、必要に応じてそれぞれをべき乗剰余計算部 332 に供給する。

【0118】

データ記憶部 (0) 333 には、法 q 上の k 個の分割秘密鍵を使用したべき乗剰余計算の計算結果 S_q が格納され、データ記憶部 (1) 334 には、法 p 上の k 個の分割秘密鍵を使用したべき乗剰余計算の計算結果 S_p が格納される。なお、S_p は、最終的に、 $M p^{d_p} \bmod p$ を与え、S_q は、最終的に、 $M q^{d_q} \bmod q$ を与えることになるものである。ここで、 $M p = M \bmod p$ 、 $M q = M \bmod q$ である。

【0119】

平文分割部 335 は、対象となる平文 M をもとに、 $M p = M \bmod p$ および $M q = M \bmod q$ を生成し、データ記憶部 (0) 333 の初期値として M_q を供給するとともに、データ記憶部 (1) 334 の初期値として M_p を供給する。

【0120】

べき乗剰余計算部 332 は、分割鍵記憶部 331 から供給される鍵インデックス値 t [i] が 1 ならば、データ記憶部 (1) 334 から前回までの計算結果 S_p (ただし、初回は初期値 $S p = M \bmod p$ となる) を呼び出して、法 p 上の分割秘密鍵 d_p [j] を使って $S p = S p^{d_p[j]} \bmod p$ の計算を行い、その結果 S_p をデータ記憶部 (1) 334 に格納する。他方、鍵インデックス値 t [i] が 0 ならば、データ記憶部 (0) 333 から前回までの計算結果 S_q (ただし、初回は初期値 $S q = M \bmod q$ となる) を呼び出して、法 q 上の分割秘密鍵 d_q [u] を使って $S q = S q^{d_q[u]} \bmod q$ の計算を行い、その結果 S_q をデータ記憶部 (0) 333 に格納する。

【0121】

なお、ここでは、鍵インデックス値 t [i] = 1 の場合に、法 p 上の分割秘密鍵によるべき乗剰余計算を行い、鍵インデックス値 t [i] = 0 の場合に、法 q 上の分割秘密鍵によるべき乗剰余計算を行うようにしているが、これに限定されるものではなく、例えば、鍵インデックス値 t [i] = 0 の場合に、法 p 上の分割秘密鍵によるべき乗剰余計算を行い、鍵インデックス値 t [i] = 1 の場合に、法 q 上の分割秘密鍵によるべき乗剰余計算を行うようにしてもよい。

【0122】

CRT 合成部 336 は、データ記憶部 (0) 333 の最終結果 S_q とデータ記憶部 (1) 334 の最終結果 S_p とを $S = \{ S p * q * (q^{-1} \bmod p) + S q * p * (p^{-1} \bmod q) \} \bmod n$ なる処理で合成し、合成結果 S (= $M^d \bmod n$) を出力部 337 に供給する。

【0123】

出力部 337 は、最終署名結果として $S (= M^d \bmod n)$ を出力する。

【0124】

以下、中国人剰余定理を使用する場合の分割秘密鍵を用いた RSA 署名手順について説明する。

【0125】

図 8 に、この場合の処理手順の一例を示す。

【0126】

なお、ここでは、法 p 乗の分割秘密鍵 d_p [j] は、j の値の小さい方から逐次使用し、法 q 乗の分割秘密鍵 d_q [u] は、u の値の小さい方から逐次使用するものとした例を示している。

【0127】

最初に、データ記憶部 (0) 333 およびデータ記憶部 (1) 334 にそれぞれ初期値として平文分割部 335 から供給される M_q 及び M_p を書き込み、また、i = 0、j = 0、u = 0 とする (ステップ S31)。

【0128】

10

20

30

40

50

以降は、 $k_p + k_q = 2k$ 回の繰り返し処理となる。

【0129】

まず、べき乗剰余計算部332は、鍵インデックス値 $t[i]$ を参照し、 $t[i] = 1$ ならば（ステップS32）、データ記憶部(1)334から S_p を読み出し、法 p 上の分割秘密鍵 $d_p[j]$ を使って、 $S_p = S_p \cdot d_p[j] \bmod p$ の計算を行い、その結果 S_p をデータ記憶部(1)334に格納し（ステップS33）、 j および i をそれぞれ1ずつインクリメントする（ステップS34、S37）。

【0130】

他方、鍵インデックス値 $t[i]$ が0ならば（ステップS32）、データ記憶部(0)333から S_q を読み出し、法 q 上の分割秘密鍵 $d_q[u]$ を使って、 $S_q = S_q \cdot d_q[u] \bmod q$ の計算を行い、その結果 S_q をデータ記憶部(0)333に格納し（ステップS35）、 u および i をそれぞれ1ずつインクリメントする（ステップS36、S37）。

【0131】

上記の2系統のうちの鍵インデックス値 $t[i]$ に応じたいずれかが行われた後に、 i の値を参照し、 i の値が法 p 上の分割秘密鍵と法 q 上の分割秘密鍵の総数 $2k$ を超えていないなら（ステップS38）、ステップS32に戻り、次のループの処理を行う。

【0132】

最終的に、ステップS38において、 $i > 2k$ となったならば、処理ループを抜けて、CRT合成部336により、データ記憶部(0)333の最終結果 S_q とデータ記憶部(1)334の最終結果 S_p とを $S = \{S_p \cdot q \cdot (q^{-1} \bmod p) + S_q \cdot p \cdot (p^{-1} \bmod q)\} \bmod n$ なる処理で合成して、平文 M に対する署名文 $S = M \cdot d \bmod n$ を求める（ステップS39）。

【0133】

以上のように処理を行うことで、中国人剰余定理を使用する場合にも、法 p 上の分割秘密鍵によるべき乗剰余計算と法 q 上の分割秘密鍵によるべき乗剰余計算とをランダムな順序に実行させて、電力解析攻撃を困難にさせる。

【0134】

このように本実施形態においては、電力解析攻撃によってRSA署名処理中の消費電力波形から、たとえ分割秘密鍵の全ての値が分かったとしても、法 p 上の分割秘密鍵と法 q 上の分割秘密鍵との区別が特定されなければ、秘密鍵 d_p および d_q を特定することができない。したがって、秘密鍵を求めるために必要な総当り数（測定される波形の個数 $k_p + k_q$ から法 p 上の分割秘密鍵の個数 k_p を選び出す組合せ）を大きくすることで、計算量的に攻撃を困難にすることができる。これまで説明した構成例では、鍵の分割数 k_p や k_q を多くすればするほど、セキュリティ強度が上がる。

【0135】

ちなみに、中国人剰余定理を使用して、秘密鍵 d_p 、 d_q を64個（上記の例でいうと $k_p = k_q = k = 32$ ）に分割した場合は、約2の60乗の鍵の組み合わせを考慮しなければ、秘密鍵は求まらないことになる。また、これに加えて、後述するように、例えば鍵の分割方法に関する情報なども可変とすると、その情報も必要となってくるため、秘密鍵の情報を得ることは非常に困難となる。

【0136】

以下では、本実施形態の変形例について説明する。

【0137】

上記では、法 p 上の分割秘密鍵 $d_p[j]$ は、 j の値の小さい方から逐次使用し、法 q 上の分割秘密鍵 $d_q[u]$ は、 u の値の小さい方から逐次使用するものとしたが、もちろん、これらとは異なる順番で鍵を使用していてもよい。

【0138】

また、上記では、法 p 上の分割秘密鍵 $d_p[j]$ の使用順序および法 q 上の分割秘密鍵 $d_q[u]$ の使用順序は固定のものであったが、法 p 上の分割秘密鍵の使用順序と法 q 上の

分割秘密鍵の使用順序の一方又は双方を、所定のタイミングで（例えば、毎回）、ランダムに変えるようにしてもよい。このように鍵の使用順序をランダムにすることによって、さらにセキュリティ強度を上げることができる。

【0139】

また、上記では、秘密鍵 d_p や d_q の分割数 k_p や k_q は固定であったが、分割数 k_p と k_q の一方又は双方を可変としてもよい。このように分割数を可変とすることによって、さらにセキュリティ強度を上げることができる。

【0140】

また、上記では、秘密鍵 d_p や d_q の分割方法として、乗法のみものを使用した。他の分割方法、例えば、加法のみ、減法のみ、除法のみの分割方法も可能であり、また、加法・減法・乗法・除法の適当な組合せも可能である。

10

【0141】

また、上記では、秘密鍵 d_p や d_q の分割方法は特定のものに固定されていたが、例えば、装置（あるいはユーザ）毎に秘密鍵の分割方法を変える（同一の装置（あるいはユーザ）では秘密鍵の分割方法は固定する）ようにしてもよいし、また、例えば、同一の装置（あるいはユーザ）についても、所定のタイミングで（例えば、毎回）、鍵の分割方法を変えるようにしてもよい。このように秘密鍵の分割方法をランダムにすることによって、さらにセキュリティ強度を上げることができる。

【0142】

また、秘密鍵 d_p と秘密鍵 d_q とで、秘密鍵の分割方法を同一にしてもよいし、異なる秘密鍵の分割方法を用いるようにしてもよい。

20

【0143】

また、図6は、分割鍵生成装置とRSA署名装置とが分離している分離型の構成をとる場合の一例であったが、分割鍵生成装置とRSA署名装置が一体化している一体型の構成をとることも可能である。図9に、この場合の暗号処理システムの構成例を示す。図9に示されるように、本暗号処理システムは、図6の外部鍵発行機310と同様の外部鍵発行機410と、図6の分割鍵生成装置320の構成要素とRSA署名装置330の構成要素を包含するRSA署名装置（べき乗剰余計算装置）420を備えている。なお、図9の構成例の動作は、基本的には、図6の構成例と同様である。図6や図9の構成例に関する利用形態の例については、第1の実施形態で図1および図4を参照しながら説明したものと同様である。

30

【0144】

また、なお、図6や図9とは異なる構成例も可能である。例えば、図6において、乱数生成部322をRSA署名装置330側に備える構成も可能である。

【0145】

ところで、これまで図6や図9を参照しながら説明してきた構成例において、さらに第1の実施形態のように、ダミー鍵を法 p 上の分割秘密鍵、法 q 上の分割秘密鍵に混ぜ込んで処理をするようにしてもよい。

【0146】

ダミー鍵を使用しない場合には、図6のRSA署名装置330や図9のRSA署名装置420において、データ記憶部(0)およびデータ記憶部(1)に加えて、ダミー鍵を使用したべき乗剰余計算の計算結果（すなわちダミーの計算結果） S' を格納するための第3のデータ記憶部（以下、データ記憶部(2)と記述する）をさらに設ける。図6のRSA署名装置330においてさらにデータ記憶部(2)338を設けた様子を図10に示す。

40

【0147】

なお、鍵インデックス $t[i]$ の代わりに、2ビット単位の鍵インデックス $tt[i]$ を考え、例えば、 $tt[i] = "01"$ の場合には、法 p 上の分割秘密鍵によるべき乗剰余計算の処理を行い、 $tt[i] = "00"$ の場合には、法 q 上の分割秘密鍵によるべき乗剰余計算の処理を行い、 $tt[i] = "10"$ or $"11"$ の場合には、ダミー鍵によるべき乗剰余計算の処理を行うようにする。なお、鍵インデックス $t[i]$ がどのような値

50

のときに、3つのうちのいずれのべき乗剰余計算の処理を行うかについての対応は、上記の例に限定されるものではなく、他の対応でも構わない。

【0148】

また、ダミー鍵を使用する場合には、図7の分割秘密鍵生成手順については、ステップS25以前の適当なタイミングにx個のダミー鍵 $du[u]$ ($0 \leq u \leq x-1$)を生成するステップを設ける。また、ステップS23では、 $(kp+kq+x-2)*2$ ビット長で且つ上記の例の場合には“01”なる2ビットを $kp-1$ 個含み且つ上記の例の場合には“00”なる2ビットを $kq-1$ 個含む鍵インデックス $tt[i]$ ($2 \leq i \leq kp+kq+x-1$)をランダムに生成する。また、ステップS24では、 $tt[0] = “01”$ 、 $tt[1] = “00”$ とするか、あるいは、その逆に、 $tt[0] = “00”$ 、 $tt[1] = “01”$ とする。 10

【0149】

また、図8のRSA署名手順は、図11のように修正すればよい。図11は、図8の手順と比較して、鍵インデックス $t[i]$ の値(2ビット)で、3系統に分岐する。すなわち、ダミー鍵によるべき乗剰余計算の系統が加えられている。なお、ステップS40では、 $tt[i] = “10”$ のときに $S' = Sp^{du[u]} \bmod p$ を行い、 $tt[i] = “11”$ のときに $S' = Sq^{du[u]} \bmod q$ を行うようにしているが、 $tt[i]$ の値にかかわらずに、 $S' = Sp^{du[u]} \bmod p$ または $S' = Sq^{du[u]} \bmod q$ の一方のみを行うようにしてもよいし、 $S' = Sp^{du[u]} \bmod p$ または $S' = Sq^{du[u]} \bmod q$ 以外の処理を行う構成も可能である。 20

【0150】

なお、ダミー鍵を使用するか使用しないかを、所定のタイミングで(例えば、毎回)、選択できる構成にしてもよい。

【0151】

また、常にダミー鍵を使用しない構成にする場合には、図6や図9の乱数生成部については、ダミー鍵および鍵インデックスを生成する機能が不要になり、ダミーの計算結果 S' を保持するデータ記憶部(2)は、それ自体が不要になる。

【0152】

なお、第1の実施形態の構成と第2の実施形態の構成とを両方備え、中国人剰余定理を使用するか使用しないかを選択可能にしてもよい。 30

【0153】

さて、これまでは、RSA暗号の署名生成の場合を例にとって説明したが、もちろん、RSAの復号の場合についても同様に可能である。この場合には、これまで説明してきたRSA署名装置がそのままRSA復号装置となる。その際、入力を暗号文 $M^e \bmod n$ とすればよく、出力としては平文 $M (= (M^e \bmod n)^d \bmod n)$ が得られる。

【0154】

また、本実施形態では、公開鍵暗号方式の一例としてRSA暗号の場合を例にとって説明したが、本発明は、その他の公開鍵暗号方式(楕円曲線暗号、DH鍵共有、ElGamal暗号等)にも適用可能である。 40

【0155】

なお、本実施形態の各々の装置は、ハードウェアとしてもソフトウェアとして実現可能である。

【0156】

また、本実施形態の各々の装置は、コンピュータに所定の手段を実行させるための(あるいはコンピュータを所定の手段として機能させるための、あるいはコンピュータに所定の機能を実現させるための)プログラムとして実施することもでき、該プログラムを記録したコンピュータ読取り可能な記録媒体として実施することもできる。

【0157】

また、本実施形態の各々の装置は、独立した装置としても、例えば計算機等への組み込み用の装置としても、実施可能である。また、例えば、本実施形態の装置をＩＣカードで実現した場合に、該ＩＣカードをＣＰＵ及びメモリ内蔵のものとし、ＩＣカードに組み込むソフトウェアとして実現することも可能である。

【０１５８】

その他にも種々の実現形態が可能である。

【０１５９】

なお、この発明の実施の形態で例示した構成は一例であって、それ以外の構成を排除する趣旨のものではなく、例示した構成の一部を他のもので置き換えたり、例示した構成の一部を省いたり、例示した構成に別の機能あるいは要素を付加したり、それらを組み合わせたりすることなどによって得られる別の構成も可能である。また、例示した構成と論理的に等価な別の構成、例示した構成と論理的に等価な部分を含む別の構成、例示した構成の要部と論理的に等価な別の構成なども可能である。また、例示した構成と同一もしくは類似の目的を達成する別の構成、例示した構成と同一もしくは類似の効果を奏する別の構成なども可能である。

10

また、この発明の実施の形態で例示した各種構成部分についての各種バリエーションは、適宜組み合わせて実施することが可能である。

また、この発明の実施の形態は、個別装置としての発明、関連を持つ２以上の装置についての発明、システム全体としての発明、個別装置内部の構成部分についての発明、またはそれらに対応する方法の発明等、種々の観点、段階、概念またはカテゴリに係る発明を包

20

含・内在するものである。従って、この発明の実施の形態に開示した内容からは、例示した構成に限定されることなく発明を抽出することができるものである。

【０１６０】

本発明は、上述した実施の形態に限定されるものではなく、その技術的範囲において種々変形して実施することができる。

【０１６１】

【発明の効果】

本発明によれば、公開鍵暗号方式に対する電力解析攻撃による秘密鍵漏洩を防ぐことができる。

30

【図面の簡単な説明】

【図１】本発明の第１の実施形態に係る暗号処理システムの構成例を示す図

【図２】同実施形態に係る分割秘密鍵生成手順の一例を示すフローチャート

【図３】同実施形態に係る分割秘密鍵を用いたＲＳＡ署名手順の一例を示すフローチャート

【図４】同実施形態に係る暗号処理システムの他の構成例を示す図

【図５】同実施形態に係る分割秘密鍵を用いたＲＳＡ署名手順の他の例を示すフローチャート

【図６】本発明の第２の実施形態に係る暗号処理システムの構成例を示す図

【図７】同実施形態に係る分割秘密鍵生成手順の一例を示すフローチャート

40

【図８】同実施形態に係る分割秘密鍵を用いたＲＳＡ署名手順の一例を示すフローチャート

【図９】同実施形態に係る暗号処理システムの他の構成例を示す図

【図１０】同実施形態に係る暗号処理システムのさらに他の構成例を示す図

【図１１】同実施形態に係る分割秘密鍵を用いたＲＳＡ署名手順の他の例を示すフローチャート

【図１２】square-multiply methodの処理手順を示すフローチャート

【符号の説明】

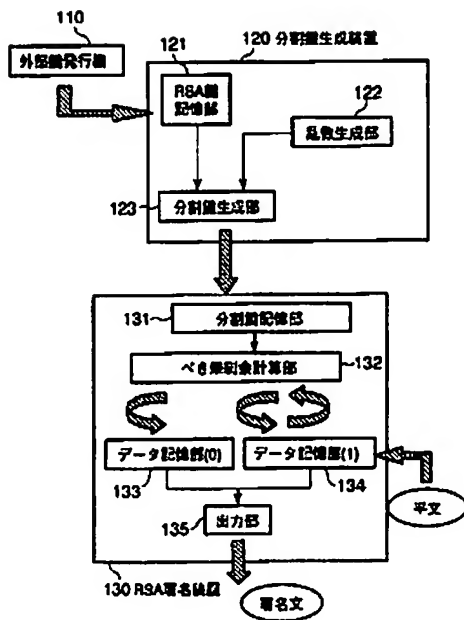
１１０、２１０、３１０、４１０…外部鍵発行機

50

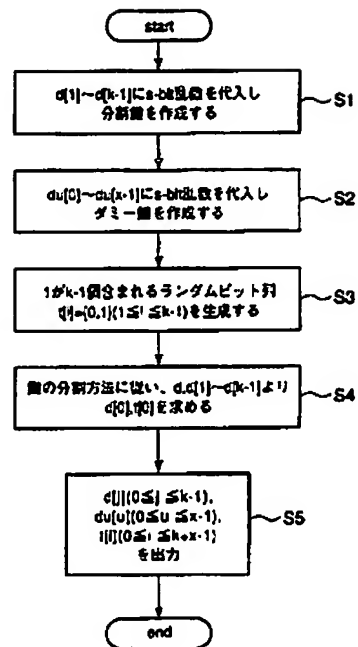
1 2 0, 3 2 0 … 分割鍵生成装置
 1 3 0, 2 2 0, 3 3 0, 4 2 0 … R S A 署名装置
 1 2 1, 2 2 1, 3 2 1, 4 2 1 … R S A 鍵記憶部
 1 2 2, 2 2 2, 3 2 2, 4 2 2 … 乱数生成部
 1 2 3, 2 2 3, 3 2 3, 4 2 3 … 分割鍵生成部
 1 3 1, 2 2 4, 3 3 1, 4 2 4 … 分割鍵記憶部
 1 3 2, 2 2 5, 3 3 2, 4 2 5 … べき乗剰余計算部
 1 3 3, 1 3 4, 2 2 6, 2 2 7, 3 3 3, 3 3 4, 3 3 8, 4 2 6, 4 2 7 … データ記憶部
 1 3 5, 2 2 8, 3 3 7, 4 3 0 … 出力部
 3 3 5, 4 2 8 … 平文分割部
 3 3 6, 4 2 9 … C R T 合成部

10

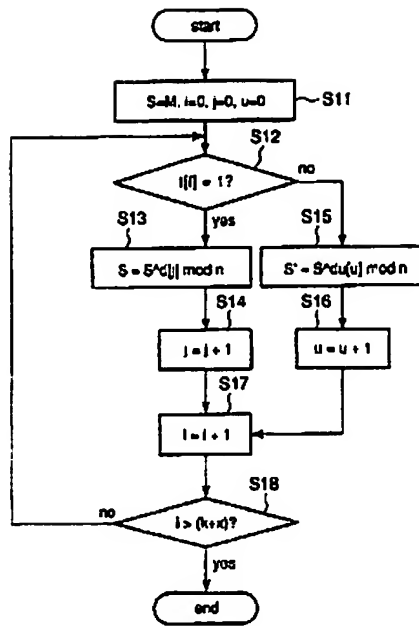
【図 1】



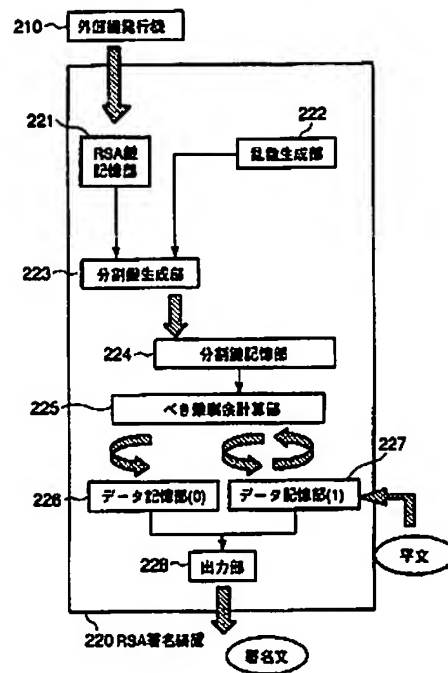
【図 2】



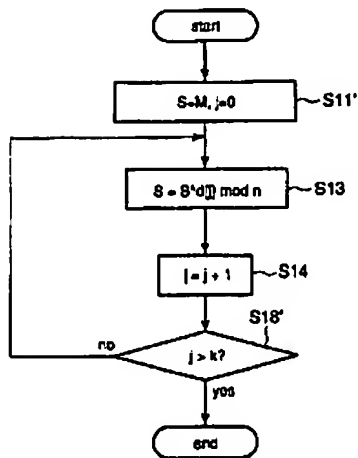
【図 3】



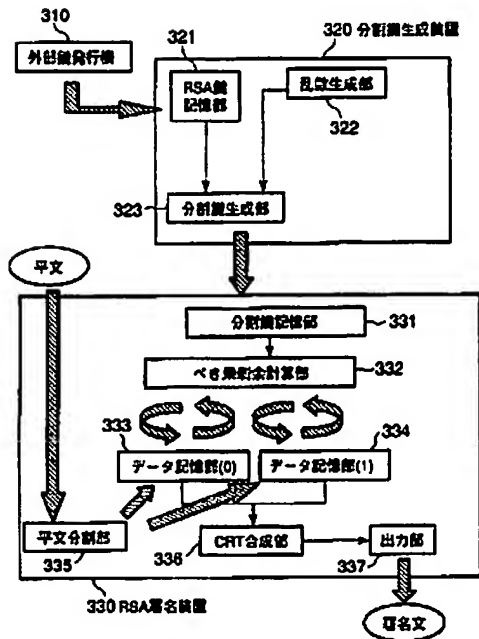
【図 4】



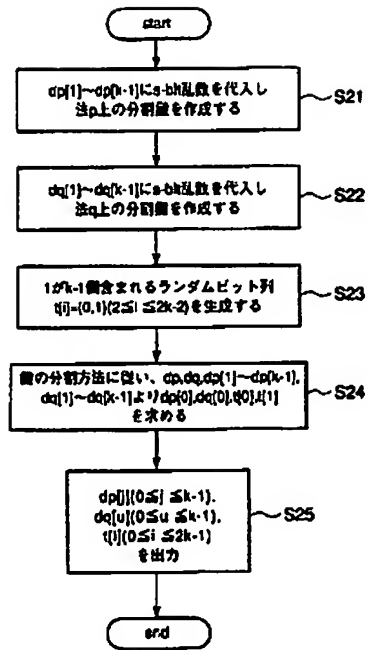
【図 5】



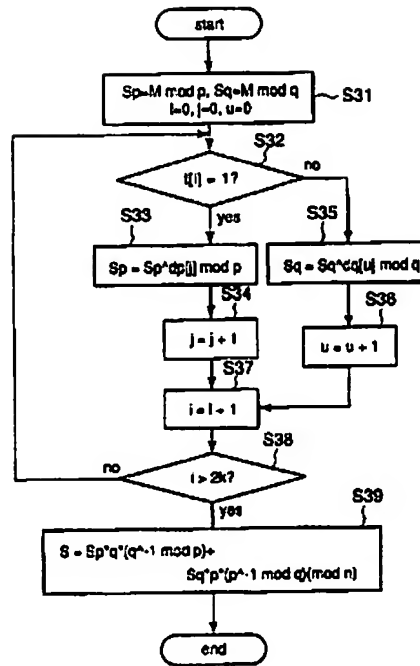
【図 6】



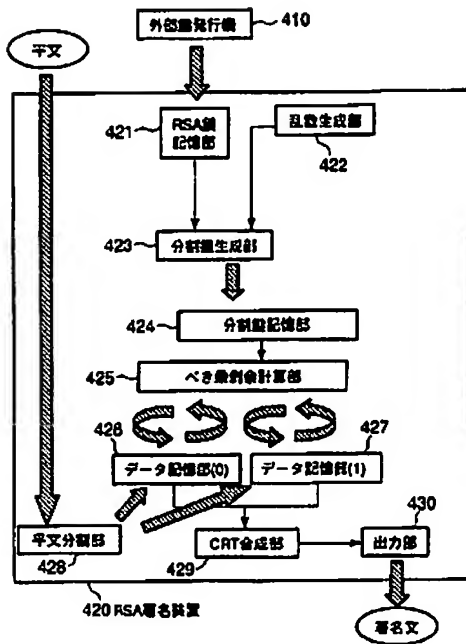
【図 7】



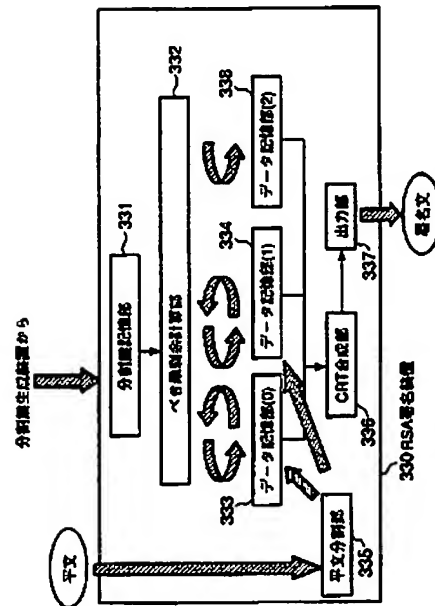
【図 8】



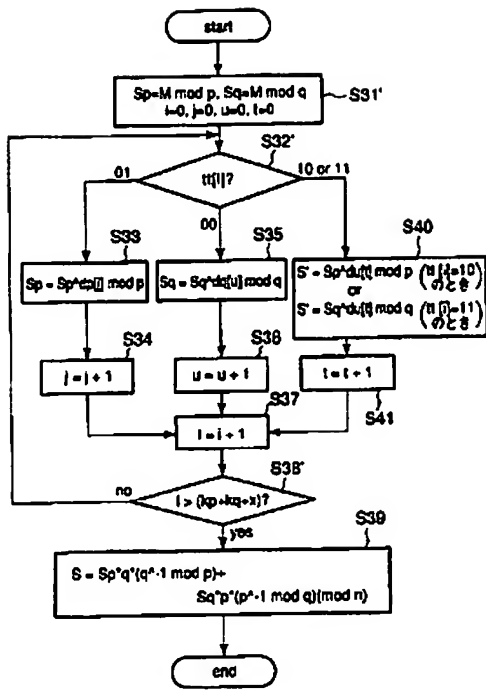
【図 9】



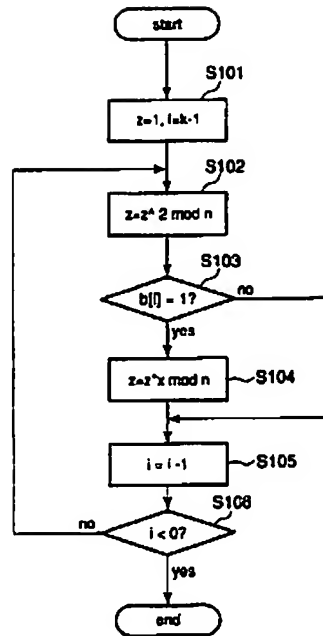
【図 10】



【図 11】



【図 12】



フロントページの続き

(74)代理人 100070437

弁理士 河井 将次

(72)発明者 友枝 裕樹

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(72)発明者 川村 信一

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

(72)発明者 新保 淳

神奈川県川崎市幸区小向東芝町1番地 株式会社東芝研究開発センター内

Fターム(参考) 5J104 AA22 EA28 JA23 NA02 NA18